Capture-Recapture – Models, Methods, and the Reality

Jens-Peder Ekros[1] jenek@ikp.liu.se
Anders Subotic[2] andsu@ida.liu.se
Bo Bergman[1] bober@ikp.liu.se

[1]Division of Quality Technology and Management

[2]Department of Computer and Information Science
Applied Software Engineering Laboratory

[1,2]Linköping University,
SE-581 83 Linköping, Sweden

Abstract

Software inspections are widely used for defect detection, and are capable of detecting defects early in development. In order to avoid spending too much resource and to assure that the inspected product has the demanded quality, a method to estimate product quality and inspection performance would be helpful. For this purpose, capture-recapture methods have been suggested. In this paper, we explore the relation between models underlying capture-recapture methods and inspection data. We have tested three hypotheses that underlie commonly used capture-recapture methods: Inspectors find the same number of defects; Defects are equally easy to detect; and, Inspectors find the same defects. We find no support for any of the three hypotheses. The paper also contributes to research by describing methods for testing the hypotheses. It is not wise to generalise from these results, as the sample analysed is small. Nevertheless, the results imply that the underlying models, or assumptions, of commonly used capture-recapture methods are not generally applicable to software engineering.

Introduction

Software plays an important role in today's society. The high dependence on software has put focus on software quality engineering. Customer satisfaction through good quality gives competitive advantage. Further, lack of quality costs, especially if quality deficiencies remain undetected from early phases of development. The lowest level of quality engineering is the detection of defects for the sole purpose of correction. The second level is quality assurance, where product measurements is compared to standards so as to assure that the shipped product is of the "right" quality. To assure is more demanding than to detect, and requires models of product quality.

Software inspection is a family of widely used methods for defect detection, capable of detecting defects early in development. In many organisations that develop software, inspections are an essential part of the process. It has been recognised that inspections have a positive effect on product quality as well as the efficiency of the development process. However, inspections demand time and resources. Preparations must be made before the inspection meeting where many key persons will attend. This is a problem. In order to avoid spending too much resource and to assure that the inspected product has the demanded quality, a method to estimate the performance of the inspection would be helpful.

Several approaches based upon different statistical techniques have been evaluated in order to get better basis to assess the above mentioned aspects. Briand et al. (1997) described three approaches:

1. Comparing inspection results with historical defect count.
2. Comparing inspection results with a baseline for defect density.
3. Estimating the number of residual defects using the current inspection results.

This paper addresses issues related to the third approach.

Lately, capture-recapture methods have gained increased attention in the software engineering community, see e.g. Eick et al. (1992). The purpose of capture-recapture methods is to estimate the size of populations.

These methods have their origin in the biological research society, recent examples include Chao (1988), Chao et al. (1992), and Pollock (1991). The methods have also been used outside of the biology area. For example, Efron and Thisted (1976) estimated the number of words known by Shakespeare.
Adapting methods to new areas, i.e. software inspections, may lead to difficulties which one has to have in mind. The most important aspect of adaptation is that of the underlying models. The different estimation methods assume certain conditions on the data, i.e. specific models. If the model does not correspond to the data, the method may give results that are either incorrect or easily misinterpreted.

From related work we have found that there sometimes is a lack of distinction between models and methods, also known as estimators, or the issue is not mentioned altogether. The bulk of work on capture-recapture in the field of software engineering has been concerned with methods, e.g. Vander Wiel and Votta (1993), Wohlin et al. (1995), Briand et al. (1997), Miller 1998, and Wohlin and Runesson (1998). Assumptions have been made, and sometimes claimed, with little support from literature or analyses, e.g. Vander Wiel and Votta (1993) and Wohlin et al. (1995). There are few tests of consequences of broken model assumptions on results, e.g. Vander Wiel and Votta (1993). However, analyses investigating how the assumptions behind methods correspond to reality are missing.

In this paper we examine published inspection data sets in order to learn more about inspector capability, defect detectability, and how these relate to each other. The importance of this work is that it provides a basis for use of prediction methods, by validating, or invalidating, model assumptions demanded by methods. In order to manage this, new variants of statistical tools have been used. The differences between inspection data and that of other fields, e.g. a low density of information in the tables, increases the difficulty of conducting tests on inspection data. One problem is that ordinary distributions cannot be used. This forces the creation of specific distributions for each specific case. These distributions depend on the size of the table, the number of rows and columns, as well as the density of the table, i.e. the ratio of ones. A number of extended computer simulations of Monte Carlo type and enumeration were conducted in order to create these distributions.

Background

Generally, capture-recapture based estimation of population size begins with sampling of the population. The results of sampling are used as parameters in an estimator function, which gives the size of the population, if certain conditions are fulfilled. In previous published work in the field of software engineering, the main focus was investigation of the performance of different methods, or estimators. In fact, methods and models are often confused. In this paper, an estimator, or method, denotes the way in which an estimate is computed. A model represents a set of assumptions on input data under which a method has been designed to work.

The most common families of models are:

1. $p_{ij} = p$: the probability of an inspector having detected a defect is constant and does not vary with inspector or defect.
2. $p_{ij} = p_i$: the probability depends on the difficulty of the defect, which varies between defects, and all inspectors have the same capability of detecting a specific defect.
3. $p_{ij} = p_j$: the probability depends on the capability of the inspector, which varies between inspectors, and all defects are equally difficult to detect for a given inspector.
4. $p_{ij} = p_i\, p_j$: the probability depends on the capability of the inspector as well as the difficulty of the defect, which both vary.
5. $p_{ij} = p_{ij}$: the detection probability might be individual depending on both inspector and defect.

Miller (1998) gave additional assumptions that relate to the process of (re-)capturing.

The above mentioned models are implicitly used in estimators. The most common estimators are Jack-knife, Maximum-likelihood, and the Chao estimator, of which there are several versions. The Jack-knife estimator is based on model number two, Maximum-likelihood on number three, and Chao estimators exist for numbers two to four.

Vander Wiel and Votta (1993) studied "the effects of broken [model] assumptions on" the Jack-knife and Maximum-likelihood estimators. The Maximum-likelihood estimator was found to perform better than Jack-knife, especially if defects were grouped to achieve homogenous detectability. Wohlin et al. (1995) claimed that the assumptions of the Jack-knife method do not correspond to reality and rejected it in favour of the Maximum-likelihood method. The claim was not supported by a test. They also evaluated a filtering technique to improve estimates of residual defects. The new method was evaluated using data from an experiment where a single document was inspected. Briand et al. (1997) examined the sensitivity of methods with respect to the number of samples used, i.e. number of inspectors. They recommended that at least four inspectors be used, and that the Jack-knife estimator was the best for four or five inspectors. The Chao estimator with the same model as Jack-knife was comparable for five inspectors, but behaved badly for four inspectors. The evaluation criticised by Miller (1998) with respect to choice of inspectors and number of data points. Wohlin and Runesson (1998) proposed two new estimation methods based on extrapolation of fitted curves. The methods are based on a number of assumptions that are not tested. The methods were evaluated with inspection data from two experiments, where the choice of artefacts was criticised by Miller (1998). Miller (1998) arrived roughly at the same conclusions as Briand et al. (1997). However, Miller recommended Jack-knife for three to five inspectors and for six inspectors both Jack-knife and the Chao estimator for model number four. In conclusion, there are no known examples in software engineering literature where the viability of the models assumed by capture-recapture methods is tested.

Models and Tests

By using capture-recapture methods on inspection data we want to predict, or assess, the remaining number of defects in the inspected document, the performance of the inspection, or both. To facilitate analysis of empirical data from a certain perspective, models that faithfully represent the data are needed. These models are the basis for analysis methods. That is, the " mathematical model … relates the attributes to be predicted to some other attributes that we can measure now. " (Fenton and Pfleeger 1996)

A number of model assumptions have implicitly been made when a method has been chosen. The methods are dependent on the underlying models that supposedly describe the data to be analysed. This is important but often forgotten. In this section we will present ways to determine model characteristics of inspection data. This gives a better basis for choosing or creating suitable estimation methods. By looking at published results from a number of inspections some conclusions regarding the underlying models have been made. We have also made a contribution in the methods to determine these models.

The type of defect-inspector table used throughout this paper is shown in Figure 1. The table represents the defects found as *r* rows and inspectors that found the defects as *c* columns. Let n_{ij} be the contents of a cell representing the detection of defect *i* by inspector *j*. If the defect on row *i* was detected by inspector *j*, n_{ij} is one, otherwise n_{ij} is zero. The number of inspectors that detected defect *i* is the number of ones in row *i*, the row sum, denoted $n_{i.}$. The number of defects detected by inspector *j* is the number of ones in column j, the column sum, denoted $n_{.j}$. The total number of detections is denoted $n_{..}$.

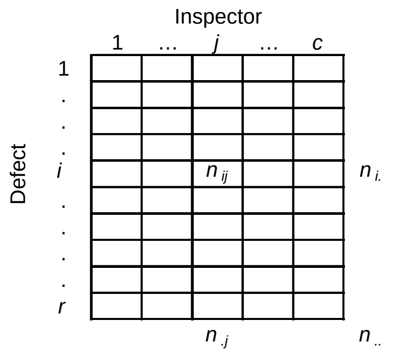

Figure 1. Graphical description of an inspection data matrix.

In the rest of this section we describe tests for analysing inspection data, mainly with respect to aspects that are relevant to the most commonly used capture-recapture methods. Interesting aspects of inspection data relate to inspectors, defects, and their relation.

Inspector Capability and Defect Detectability

The assumptions regarding the capabilities of inspectors are concerned with the number of defects detected by each inspector. Intuition often suggests that inspectors have different capabilities but this has not yet been tested, see e.g. Wohlin et al. (1995). The capability of inspectors is tested by comparing the number of defects found by each inspector with the average number of defects detected.

A test statistic similar to the Chi-square test (Everitt 1992) is utilised, $Q = \sum_{j=1}^{c} \left(n_{.j} - \frac{n_{..}}{c} \right)^2 \Big/ \frac{n_{..}}{c}$. Problems arise when the expected values of row or column sums are too small. The Chi-square test is commonly considered to work less than well for values below five. With respect to row sums, this criterion is not fulfilled by any of the tables analysed in this paper. The expected values of column sums fulfil the criterion for roughly half of the tables. Preferably, under the null hypothesis of equality, the statistic for an observed table is compared with a reference distribution. Since the Chi-square distribution is not applicable, a substitute distribution has to be constructed.

Enumeration was the approach chosen for creating the distribution of the Q-statistic. That is, in e.g. analysing inspector capability, the set of column sums of a table is analysed. The total number of detected defects, the number of 1:s in a table, are distributed in all unique ways over the columns. The Q-statistic for each permutation is computed and its occurrence is weighted by its probability under the assumption of homogeneity. The result is a reference distribution adapted to characteristics of the table. The p-value for the Q-statistic of the observed table, Q_o, is obtained from the distribution as $p = P(Q > Q_o)$, where the variable Q belongs to the distribution.

Similarly to inspector capability, defect detectability is defined as the fraction of inspectors that found a specific defect. That is, a defect found by many inspectors is said to have a higher degree of detectability. Wohlin et al. (1995) recommended that defects be divided into two groups based on the number of inspectors that found each defect. When only one inspector found a defect the defect was put in a low detectability group. Defects found by more than one inspector were put in a high detectability group. However, this reasoning hides the assumption that defects have different probability of detection. The test for difference among defects is the same as for inspector but along the other dimension of the table.

Defects and Inspectors Combined

A third approach to test inspection data is to consider defect and inspector characteristics at the same time. That is, the test helps determine if inspectors are equally good in finding different types of defects. Since the cell values of the tables analysed are either one or zero, chi-square tests are not appropriate. An alternative test statistic is proposed. The new test statistic is built up by a sum of the differences between the inspectors based on every specific defect.

Let *l* and *m* be the identities of two inspectors. Let n_{ij} be one if defect *i* was detected by inspector *j*, and zero otherwise. A matrix *K* is created where $k_{lm} = \{number\ of\ \ n_{il} > n_{im}\}$ where $l \neq m$ and $k_{ll} = \{number\ of\ \ n_{il} > 0\}$. A proposed test statistic is $T = \sum_{l} \sum_{m} \left(k_{lm} \big/ k_{ll} \right) * w_l$ where $w_l = 1$.

The diagonal of *K*, k_{ll}, represents the number of defects found by each inspector. Values outside the diagonal i.e. k_{ij} represents the number of defects found by inspector *i* but not by inspector *j*. That is, a big number outside the diagonal indicates that one of the inspectors found many defects not found by the other. The matrix is quadratic, but it is rarely symmetrical, as e.g. inspectors seldom find the same number of defects.

The distribution of the T-statistic is not known, but needed in order to determine the meaning of a T-value. Thus, the distributions of T for each table have to be generated. Due to problem size, enumeration is not an option. Instead, Monte Carlo simulation is used. That is, the distribution of T for a given table is acquired by computing the T-statistic for n randomly generated tables, with similar characteristics as the observed table. The simulations are crucial to the reliability of the analysis, and so the bulk of work has been spent on trying to achieve a simulation that represents the true probability distribution. Again, the rationale for simulations is that the distribution of the test statistic is unknown.

Results

In this section we investigate characteristics of published inspection data sets and in the process we provide ways of testing model assumptions. The data sets were mainly taken from Freimut (1997), where the majority originates from experiments using NASA subjects. A data set from Wohlin et al. (1995) and one from Myers (1978) were also used in some of the analyses. The results are used for accepting or rejecting the hypotheses stated above. Three different hypotheses are tested.

Inspector homogeneity

For each table *k* the test-variable Q_k is calculated. The value Q_k is compared to an empirical distribution constructed using enumeration, as described above. For this analysis, two data sets were not used due to time and size complexity problems, as there were many combinations to enumerate.

Under the null hypothesis of homogeneity, the expected value of p is 0.5. A low p-value is the result of large differences between the number of defects discovered by different inspectors. A single small p-value supports rejection of the null hypothesis but is not enough to safely reject it. By combining analyses of a number of tables we get a greater body of evidence. Figure 2 shows the p-values for the 22 different data tables tested. Under the null hypothesis the p-values would be evenly distributed between 0 and 1. This does not seem to be the case. The conclusion is therefore that the null hypothesis is rejected. Thus, based on this set of data, we can say that inspectors generally do not find the same number of defects, i.e. inspector capability varies with inspectors.

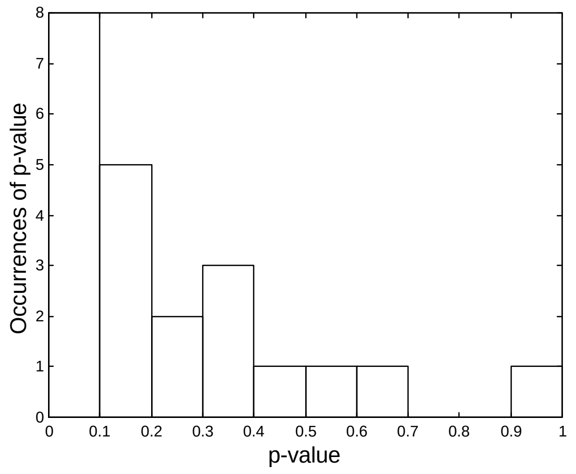

Figure 2. p-values for inspector homogeneity.

Defect homogeneity

Analysing the defect detectability in the same way as inspector capability shows that it can be concluded that defects do not have equal detectability. The p-values for the 22 data sets shown in Figure 3 clearly indicate a non-equal distribution. That is it cannot be said that the different defects generally have equal detectability. For this analysis, two data sets were not used due to time and size complexity problems, as there were many combinations to enumerate.

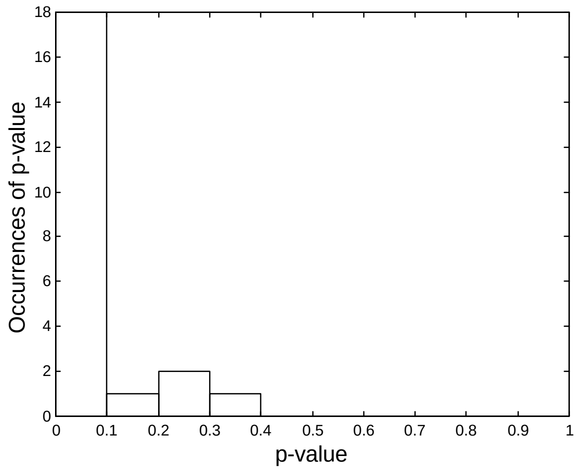

Figure 3. p-values for defect homogeneity.

Another way of representing the p-values is with a scatter plot, shown in Figure 4. Here p-values for inspector capability and defect detectability for 22 tables are plotted against each other. There are no real outliers. That is, no table plots in the upper right quarter. The plot gives stronger support for rejection of the null hypothesis for defect detectability than for inspector capability. That is, defects are more heterogeneous than inspectors. Even though inspectors seem to be more homogeneous than defects, they are predominantly heterogeneous.

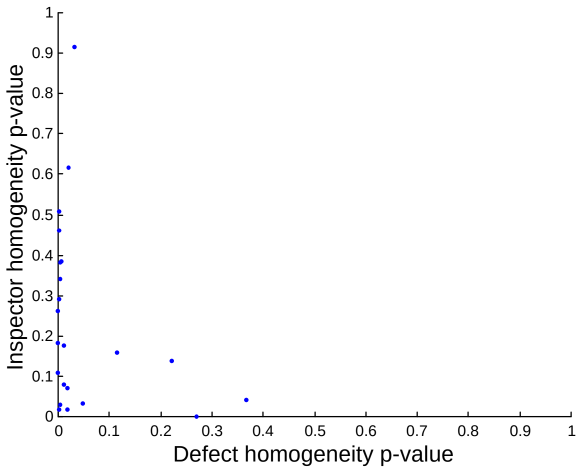

Figure 4. Scatter plot of defect and inspector homogeneity p-values.

Inspector and defect combined homogeneity

In this section, we analyse the relations between defects and inspectors. In Figure 5 the p-values from this analysis is presented. The results stem from 10.000 simulations of 24 tables. As in the other cases it can clearly be seen that the p-values are not evenly distributed between 0 and 1. A low p-value is the result of large differences between inspectors; i.e. inspectors find different defects. The distribution of p-values is skewed towards zero. In fact, 50 percent of the values are lower than 0.1 and about 85 percent are less than 0.5. This situation is highly unlikely under the assumption that inspectors find the same defects. The implication is that there is no support for “inspector profiles”, i.e. groups of inspectors that find similar subsets of the defect population. If groups of inspectors found largely the same defects p-values should be skewed toward the right. It may be harsh to reject the concept of “inspector profile” based solely on this analysis. However, analytical advocacy is no longer enough.

This is analysis differs from the test of inspector homogeneity above, where it was shown that different inspectors find different number of defects.

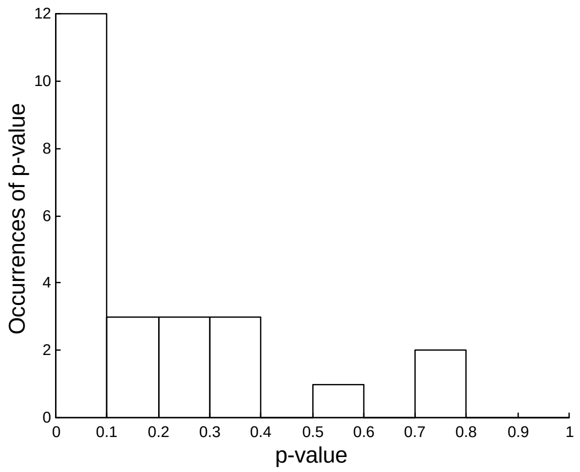

Figure 5. The p-values for combined inspector and defect homogeneity.

Conclusion and Discussion

In this paper we went back to the basics, that is, using the information from inspections to explore the underlying models that are assumed to characterise inspection data. The road towards better methods and correct use of existing methods will start from the most logical point, the distribution and properties of the data.

We have tested three hypotheses occurring in capture-recapture work in the software engineering field:
1. Inspectors find the same number of defects.
2. Defects are equally easy to detect.
3. Inspectors find the same defects.

We find no support for any of the three hypotheses. These results imply that the underlying models, or assumptions, of commonly used capture-recapture methods are not applicable to the software engineering data analysed in this paper. Even though 24 data sets were analysed, 16 of these originate from NASA. Thus, it is not wise to generalise from these results. However, the results suggest that it is wise to test model assumptions. Testing model assumptions requires good data, which in turn requires good instrumentation and collection procedures. We have instrumented the inspection process of an industry partner, and are currently awaiting data to accumulate.

By exploring data from several inspections we are able to draw general conclusions about how the data is formed. This includes measures of correlation between inspectors and between faults. Other aspects are the distributions of inspectors' performance and defect detectability. Naturally, it may not be possible to find a single specific model that will explain all relationships between inspectors and defects. The data sets used here, to estimate characteristics of product and inspection process, have only two parameters. It is not unlikely that other product, process and resource attributes could increase the usefulness of estimators. There may be important differences between instances of inspections, e.g. differing inspection rate, team expertise, type of document, and organisational culture.

Even though universal models are few and far between, the goal is to find general models. It may be easier to find methods to derive situation specific models. These methods can then be used together with local inspection data to assure that a suitable estimation method is used. Still, it is questionable how the information gained can be used. Does the estimate of defect content depend mainly on the product, the measurement process, or both? If the inspection process is unstable, measurement noise may obscure product attributes. For example, it is hard to determine if a high defect count is the result of a bad product, a good inspection, or both. An accompanying metric is needed for normalisation.

Acknowledgements

The authors wish to thank Mary Helander for her helpful comments on this paper. This work was supported by the Swedish National Board for Industrial and Technical Development (NUTEK), administrated by the Swedish Institute for Applied Mathematics, and the Swedish Foundation for Strategic Research through the ECSEL graduate school at Linköping University, Sweden.

References

Briand, L. C., Emam, K. E., Freimut, B., and Laitenberger, O. (1997). "Quantitative Evaluation of Capture-Recapture Models to Control Software Inspections." *Report 97-22*, ISERN.

Chao, A. (1988). "Estimating Animal Abundance with Capture Frequency Data." *Journal of Wildlife Management*, 52(2), 295-300.

Chao, A., Lee, S.-M., and Jeng, S.-L. (1992). "Estimating Population Size for Capture-Recapture Data When Capture Probabilities Vary by Time and Individual Animal." *Biometrics*, 1992(March), 201-216.

Efron, B., and Thisted, R. (1976). "Estimating the number of unseen species: How many words did *Biometrika*, 63(3), 435-447.

Eick, S. G., Loader, C. R., Long, M. D., Votta, L. G., and Vander Wiel, S. (1992). "Estimating Software *Proceedings of the Fourteenth International Conference of Software Engineering*, May, Melbourne.

Everitt, B. S. (1992). *The Analysis of Contingency Tables*, Chapman & Hall, London.

Fenton, N. E., and Pfleeger, S. L. (1996). *Software Metrics: A Rigorous and Practical Approach*, International Thomson Computer Press, London.

Freimut, B. (1997). "Capture-Recapture Models to Estimate Software Fault Content," *Masters Thesis*, University of Kaiserslauten.

Miller, J. (1998). "Estimating the number of remaining defects after inspection." *Report 98-24*, ISERN.

Myers, G. J. (1978). "A Controlled Experiment in Program Testing And Code Walkthroughs/Inspections." *Communications of ACM*, 21(9), 760-768.

Pollock, K. H. (1991). "Modeling Capture, Recapture and Removal Statistics for Estimation of Demographic Parameters for Fish and Wildlife Populations: Past, Present, and Future." *Journal of the American Statistical Association*, 86(413), 225-238.

Vander Wiel, S. A., and Votta, L. G. (1993). "Assessing Software designs using Capture-Recapture Methods." .

Wohlin, C., and Runesson, P. (1998). "Defect Content Estimations from Review Data." *Proceedings of the Twentieth International Conference on Software Engineering*, April, Kyoto, 400-409.

Wohlin, C., Runesson, P., and Brantestam, J. (1995). "An Experimental Evaluation of Capture-Recapture in *Software Testing, Verification and Reliability*, 5, 213-232.